The International Journal of Digital Curation
Volume 7, Issue 2 | 2012

Digital Forensics Formats: Seeking a Digital Preservation Storage Container Format for Web Archiving

Yunhyong Kim,
Humanities Advanced Technology and Information Institute,
University of Glasgow

Seamus Ross,
Faculty of Information,
University of Toronto, Canada
and
Humanities Advanced Technology and Information Institute,
University of Glasgow

Abstract

In this paper we discuss archival storage container formats from the point of view of digital curation and preservation, an aspect of preservation overlooked by most other studies. Considering established approaches to data management as our jumping off point, we selected seven container format attributes that are core to the long term accessibility of digital materials. We have labeled these core preservation attributes. These attributes are then used as evaluation criteria to compare storage container formats belonging to five common categories: formats for archiving selected content (e.g. tar, WARC), disk image formats that capture data for recovery or installation (partimage, dd raw image), these two types combined with a selected compression algorithm (e.g. tar+gzip), formats that combine packing and compression (e.g. 7-zip), and forensic file formats for data analysis in criminal investigations (e.g. aff – Advanced Forensic File format). We present a general discussion of the storage container format landscape in terms of the attributes we discuss, and make a direct comparison between the three most promising archival formats: tar, WARC, and aff. We conclude by suggesting the next steps to take the research forward and to validate the observations we have made.

International Journal of Digital Curation (2012), *7*(2), 21–39. http://dx.doi.org/10.2218/ijdc.v7i2.227

The *International Journal of Digital Curation* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. The IJDC is published by UKOLN at the University of Bath and is a publication of the Digital Curation Centre. ISSN: 1746-8256. URL: http://www.ijdc.net/

 doi:10.2218/ijdc.v7i2.227

Introduction

The selection of a storage container format for digital materials that facilitates the long-term accessibility of digital object content, and supports the continued recognition of behaviour and functionalities associated to digital objects, is one of many core tasks of a digital archive. This task is especially challenging with respect to complex aggregate digital objects, such as weblogs, involving multimedia objects that are produced in varying formats to carry out a wide range of interactive functionalities, including dynamic changes overtime, and displayed using distributed information within the context of social networks. As a first step to meet this challenge, we present here results of our preliminary investigations examining storage container formats likely to benefit a dynamic weblog archive, a study conducted as part of the BlogForever project[1], which aims to create a platform for aggregating, preserving, managing and disseminating blogs.

There have been many studies on the impact of digital object formats on the preservation of digital information (e.g. Brown, 2008; Todd, 2009; Buckley, 2008; Christensen, 2004; Fanning, 2008; McLellan, 2006). The retention of essential object properties can be facilitated by examining the preservation attributes of the file format. Some of these (e.g. scale of adoption and disclosure, support for data validation, and flexibility in embedding metadata) have surfaced elsewhere as sustainability factors (cf. Library of Congress sustainability factors[2]; Arms & Fleischhauer, 2003; Rog & van Wijk, 2008; Brown, 2008) and as factors that capture the format's capacity to retain significant digital object properties (Hedstrom & Lee, 2002; Dappert & Farquhar, 2009; Guttenbrunner et al., 2010).

Most of these studies seem to be focused on considerations of individual digital object formats and, even then, generate many differences of opinion. There has been little consensus on best practices for selecting storage container formats (e.g. tar) that aggregate or capture collections composed of multiple object types, such as we might encounter within a single standalone computer, a complex office system, or a web archiving environment. While formats such as WARC [A3][3] have been proposed and developed into an international ISO[4] standard, these recommendations are rarely based on a comparison of a range of formats using the full range of preservation attributes within the same environmental setup. Even when storage architecture is discussed on a wider scale, it often comes focused on one or two selected factors[5] (e.g. software and hardware scalability and costs).

In the following, we discuss a core set of preservation attributes for storage formats. These include those that have been addressed in common by several previous

[1] Partially funded by the European Union's Seventh Framework Programme (FP7-ICT-2009-6) under grant agreement n° 269963.

[2] Planning for Library of Congress Collections - Formats, Evaluation Factors, and Relationships: http://www.digitalpreservation.gov/formats/intro/format_eval_rel.shtml

[3] Throughout this paper, references in square brackets, expressed as a number preceded by the letter A, refer to those in the left most column of Table 1 in the appendix.

[4] ISO: http://www.iso.org (e.g. WARC, ISO 28500:2009)

[5] For example, see: http://www.digitalpreservation.gov/meetings/documents/othermeetings/Newman.pdf

studies on file formats, such as those conducted by the UK Digital Curation Centre[6] (e.g. Abrams, 2007), the US Library of Congress, and the technology watch reports published by the Digital Preservation Coalition[7] (e.g. Todd, 2009). However, we have augmented the set of attributes to reflect an increased cognizance of the concepts covering the quality and completeness of data, as reflected in the ability to represent the full digital content of an object and/or data. The central role of quality and completeness of data has been observed as a relevant factor before (e.g. Todd, 2009; Pipino, Lee &Wang, 2002; Batini & Scannapieco, 2006; Huc et al., 2004). However, the "completeness of data" we address here refers to much more than the target digital object content. For example, in the digital environment, provenance evidence surrounding digital objects can be derived from information external to the object, such as file modification dates, lists of files that were deleted, logs of processes (e.g. installation of programs) and resulting errors, and trails of programs that had been run on the system. This kind of history is retained on the system disk, as a result of often tacitly understood standard practice in software design and systems administration[8], and should be retained to track accountability (not only with respect to humans but also software and hardware). Once you reduce the preservation activity to that associated with digital objects only, all this supporting information tends to become hidden and may be even lost. Indeed, although we focus here on storage, we believe that this is a reductionist approach to preservation, and that real advances will come from system preservation and system thinking (e.g. Checkland, 1981), based on an understanding of complex systems driven by inter-related data.

We have also placed more emphasis on scalability (e.g. measured by compression ratio to meet storage requirements, and decompression speed to reduce overheads on any processes that take place on the material) and flexibility (e.g. being able to deal with multiple types, sizes, and numbers of digital materials through a variety of operating systems) than previous studies. Scalability and flexibility are crucial within the web environment where we need to support rapidly growing data, distributed processing, aggregation of multimedia objects, and sophisticated approaches to search.

In the next section, these observations will be reflected in our proposal of seven cores attributes for assessing storage container formats.We will then discuss a range of container formats with respect to these attributes, and make some concluding remarks with suggestions of next steps in the final section.

Seven Core Preservation Attributes for File Formats

We propose seven core attributes that should be considered with respect to storage container formats for the purposes of supporting digital preservation, based on current knowledge. As mentioned in the previous section, these attributes were selected to reflect preservation requirements identified through other research and application development initiatives, such as the sustainability factors for formats discussed at the Library Congress.[9]

[6] Digital Curation Centre: http://www.dcc.ac.uk

[7] Digital Preservation Coalition: http://www.dpconline.org

[8] For example, see: http://c2.com/cgi/wiki?LoggingBestPractices

[9] Planning for Library of Congress Collections – Sustainability Factors: http://www.digitalpreservation.gov/formats/sustain/sustain.shtml

 doi:10.2218/ijdc.v7i2.227

However, previous studies have placed much emphasis on front-end isolated formats for individual digital objects. The attributes here include the notion of completeness of data, intended to consider the extent of contextual metadata[10] (e.g. file system information, permissions and error logs) surrounding the object that is being captured. We also put weight on scalability, not only in terms of minimising storage and optimising management efficiency with respect to variations in the quantities of data (crucial in the case of web archives that become increasingly bigger in size and diverse with respect to included object types, or data collected from scientific instruments), but also in terms of reducing overhead with respect to sophisticated data mining and search technologies that are likely to play a more ubiquitous role in the future. The attributes are described below along with Library of Congress sustainability factors (LC SF) in parenthesis, for comparison, where relevant:

1. **Completeness of data**: The container format should preserve data as closely as possible to raw data at the time of storage or capture. For example, this could be a sector-by-sector replication (e.g. disk image) of the raw data on a system disk, block-by-block replication of tape storage, or packet-by-packet recording of streamed content as it was captured, inclusive of any file structure, dependencies, and history. This:
    i. Minimises deterioration and information loss;
    ii. Maximises the chances of preserving file system information (e.g. directory structure, file size, permissions, encoding, any relationships and dependencies between files and executables);
    iii. Increases the possibility of retaining extra information about changes that have been made on the disk to be used for tracking accountability, integrity, authenticity, and maximising recoverability.
2. **Recoverability of data**: The container format should support the recovery of data wherever possible. For example, one corrupted file or sector, if possible, should not pose serious problems in recovering other files or sectors in the archive.
3. **Support for data validation** (cf. LC SF "technical protection mechanisms"): The format should support validation procedures. For example, the container format should provide:
    i. Piecewise hashing utilities (i.e. programs that hash arbitrary sized blocks of data, such as md5deep[11]) and digital signatures to verify it as an authentic representation of the initial instantiation (Ross, 2006); and,
    ii. Optional means of encryption[12] to protect the data from mal-manipulation or illicit access.

    While these functions can be added in some cases, it is best to minimise the accumulation of functionality through the use of third-party tools and added procedures, as this increases overhead and the margin for introducing errors.

[10] Rather than *contextual*, the term *pragmatic* may be more precise, as it describes use contexts of data.

[11] Md5deep: http://md5deep.sourceforge.net/

[12] Since encryption keys are stored separately from the object itself, the use of encryption can be preservation hostile.

4. **Scalability of data management processes**: The format should have properties that make all processes within the archive scalable to handle files of any size, datasets of any size and added services. In particular, the format should:
    i. Not limit the size of input file, output file, and/or media; and,
    i. Support efficiency with respect to storage and processing speed. For example, the format should:
        - Have inherent efficient and effective compression[13] methods, which could be used to reduce storage requirements;
        - If possible, not require decompression for accessing information within the stored data (e.g. searching and indexing); and,
        - Support random access of files within the archive.

2. **Transparency** (cf. LC SF "disclosure", "impact of patents", and "transparency"): Any tools and specifications involved in the format should be a publicly published open standard and non-proprietary to avoid restrictions regarding activities that support long-term preservation and access of material in the archive, such as making modifications to the format, distributing new versions, and tracing accountability and authenticity.
3. **Flexibility of embedding metadata** (cf. LC SF "self-documentation"): The container format should, if feasible, support the possibility of embedding user-defined metadata with the data objects.
4. **Flexibility in handling data** (cf. LC SF "external dependencies"): The container format must be:
    i. Able to capture data objects in their entirety or in small portions;
    ii. Able to handle any media type (e.g. text, image, audio, video, executable);
    iii. Able to process any source of material (e.g. entire disk contents, folders, files, webpages, websites) whether it is acquired through the network or provided on some form of storage media; and,
    iv. Accessible using a variety of methods, environments and operating systems.

Comparison of Storage Formats

In this section, we compare several file formats that have been widely accepted as formats for storage of information, with respect to the attributes identified previously. A list of widely used formats is presented in Table 1, shown in the appendix.

The formats in Table 1 will be broadly considered with respect to five format categories:

[13] Compression's major drawback is that the redundancy in an object is eliminated; redundancy can be valuable when recovering damaged objects. Many studies recommend it be avoided. That said, even though we recognise compression as preservation hostile, the ubiquitous use of compression in information management would make it imprudent to ignore it.

 doi:10.2218/ijdc.v7i2.227

- Formats for archiving content, mostly intended for aggregating, storing, transferring, and backing-up the content (e.g. tar [A26], International Internet Preservation Consortium WARC [A27], AXF [A5]).
- Formats that capture raw data, including or excluding unused portions, as it is on the disk, mostly intended for recovery or installation (e.g. partimage [A20], dd raw image [A10]).
- Combinations of formats for archiving content or capturing raw data with standard compression tools (e.g. gzip [A14], zip [A27], bzip2 [A7], lzma [A18]).
- Common formats that combine archiving and compression (e.g. 7-zip [A23], SEA ARC [A4], cfs [A8], kgb [A17], PeaZip [A21]).
- Forensic disk image formats (e.g. aff [A1], aff4 [A2]).

The examples listed above are not meant to constitute an exhaustive list of storage container formats by any means. Some formats (such as the EnCase image format [A12] and other proprietary formats for forensics, and rar [A22] format for archiving content) were omitted because they are clearly restricted and closed proprietary formats. Also, formats whose license status was hard to resolve (e.g. BagIt[14] [A6]), formats which have a stable extended version (e.g. Internet Archive ARC [A3], now extended by the ISO standard WARC [A27]), and formats that are designed for limited purposes (e.g. jar [A16] for java applications and associated libraries, and iso image [A15] for optical media) have also been excluded. Formats such as cpio [A9] are not extensively discussed here.

Some formats have little documentation and support. This may be because the format is associated to a linux native command (e.g. shar [A25] and dd raw image [A10]), old (e.g. SEA ARC [A4]), and/or not widely adopted (e.g. cfs [A8] and kgb [A17]). While we have mentioned them in some of our discussion, the lack of documentation and support would suggest them to be unsuitable in a large scale preservation context. Likewise, formats for which there is no evidence of further planned development (e.g. forensic format gfzip [A13], frozen since 2006), or those tied to a specific program (e.g. sgzip [A24], native format of forensic software PyFlag) or specific platform (e.g. dmg [A11] for MAC OS X) seem unsuitable for serious consideration as candidates for preservation formats.

The container formats can first be compared on the basis of compression and decompression speed, and compression ratio, which may impact on system performance and management cost. We have excluded any discussion of compression methods, such as xz-utils [A28] and lzop [A19], which have not been adopted widely. The formats above are not accompanied by compression, and therefore actually have the best compression and decompression speed. However, they also require the largest amount of storage, which may impact on system design (and, hence, also on performance) and maintenance cost. The format tar compressed using gzip and bzip2 has been compared to 7-zip and PeaZip on the basis of compression ratio and compression speed by Nieminen (2004) who found that, while 7-zip produces the best

[14] BagIt Library (BIL) is described as Public Domain: http://www.digitalpreservation.gov/partners/resources/tools/. If this denotes Public Domain Certification, this could be invalid outside the USA.

compression ratio, tar+bzip2 and tar+gzip show the best ratio to speed comparator. Other studies that have compared the gzip, bzip2 and lzma compression methods show that, while lzma outperforms the other two in terms of compression size, gzip is significantly superior to the other two in terms of compression and decompression speed (Collins, 2005; Klausmann, 2008). The gzip compression method also has the least demanding memory requirements. While there is no information on compression ratios for WARC, or AXF in combination with bzip2, gzip, and zip, as WARC and AXF are container formats that do not make special provision to optimize size of embedded objects beyond the capability of a selected compression algorithm, it cannot be expected to greatly outperform tar (with a selected compression algorithm) in terms of compression ratio. We could not find a direct comparison of compression ratio and speed between the above formats and the forensics file format aff. However, we do know that the compression algorithms supported within aff are zlib and lzma[15]. The former has a typical compression ratio of 2:1 to 5:1[16], which is comparable to that of gzip. The latter is the compression algorithm for 7-zip. This suggests aff format's potential to compete with tar+gzip and 7-zip in terms of compression ratio and speed. Furthermore, aff has the advantage that it comes with the tools that allow the contents to be read without decompression.

Earlier, we presented a general discussion on storage container formats with respect to our seven attributes extracted from the literature. We have followed up on the discussion with a direct comparison between tar, WARC, and aff, three formats listed above that our preliminary analysis indicated to be the most promising. While AXF also claims to be an open standard conforming to preservation aims, it is a very new development. At the time of writing this paper, there was precious little documentation and source code publicly accessible, it was difficult to assess. For this reason, we propose that we should reserve judgement on this format at this stage with regards to its suitability for inclusion in large scale long-term storage initiatives.

General Discussion of File Format Attributes

In this section, we first present some broad observations on various formats with respect to several of the attributes identified earlier. We have organised these under four headings: completeness of data, recovery and validation, scalability, and flexibility. Transparency was not discussed separately, as we have opted, as evidenced throughout the paper, not to consider container formats that are not public open source, and that are not well documented.

Completeness of data

There are different degrees of information being archived in each of the formats listed. For example, tar will save systems information, such as permissions and file directory structure. Others, such as partimage, have limitations on supported file systems and do not retain information from unused sectors.[17] Formats such as 7-zip do not retain file permissions across platforms. For instance, data on a Windows system aggregated using 7-zip would lose file permission information when transferred onto a

[15] We intend to conduct such a study. See the section below covering next steps.

[16] Zlib Technical Details: http://www.zlib.net/zlib_tech.html

[17] Information residing in unused sectors may relate, for instance, to erased files, and may well be of much long-term interest and even value.

 doi:10.2218/ijdc.v7i2.227

Linux machine, as these attributes will be reset upon transfer. Many of these formats have intrinsic and implicit ways of handling processes that are not widely known, and that impact on their sustainability for preservation purposes. The inability to retain information of this sort also manifests in formats such as WARC, which is designed to aggregate resources on the Internet in a descriptive, surface-oriented fashion without much regard to original file system structure or the file system characteristics of the embedded resource (e.g. image). In contrast, forensics formats are implemented to keep the data as close to the way it was at the time of creation, as this can constitute vital evidence in judicial contexts.

Recovery and validation

Publicly available information on archive file formats (excluding WARC and AXF) show that only shar, ace, afa, arj, DGCA, WinMount format, rar, and ultra compressor II come with support for integrity checks, recovery records and encryption.[18] These formats are proprietary, poorly documented (e.g. shar) or have a limited community of support (e.g. DGCA). The WARC format, as far as we know, does not have any validation mechanisms (e.g. checksum) built into it. In contrast, forensic disk images (e.g. aff) almost always come with some means of supporting all three, as they impact the weight a court might give to the extracted information when it is produced as evidence.[19] While the Archival eXchange Format (AXF) does provide validation mechanisms, its provisions for recovery – that is, robustness against errors – are yet to be tested. In fact, while with many container formats the corruption of part of the data leads to the loss of a big chunk of data, formats like Advanced Forensics Format (aff) have provisions for the restoration of maximum amount of the uncorrupted data.

Scalability

Many of the listed formats have limitations on the size of the input and output file that they can produce. For example, older versions of tar only allowed up to a file size of eight gigabytes. The elasticity and processability of a format are key aspects of their scalability. Even some forensic file formats came with this limitation. However, unlike forensic file formats, most of the other formats do not allow easy partitioning of the data to be archived into blocks of user-defined size. In addition, newer versions of forensic file formats, such as Advanced Forensics Format (aff), have lifted the limitation on file size. More importantly, some archival formats (e.g. tar) do not allow random access to data, so for these there is no way to retrieve individual files without decompressing and unpacking everything. As a result, this will incur a significant overhead for management (e.g. migration of selected file types within the archived object), indexing, and retrieval operations within the archive. Even when a format allows random access (e.g. 7-zip), it is often the case that the selected file has to be decompressed before processing. Forensics formats, such as aff, in contrast, allows searching and analysis of the data without any decompression.

Flexibility

In terms of metadata, both WARC and AFF are designed to support user-defined metadata. The format tar and other content archival formats (partimage and dd raw

[18] Wikipedia – Comparison of archive formats: http://en.wikipedia.org/wiki/Comparison_of_archive_formats

[19] In some jurisdictions, it may even impact the admissibility of the material.

image) support only a limited amount of predefined metadata. This is natural, as content archival formats and raw disk images are generally born as a means of storing and transferring data from one location to another, while WARC and forensics formats are designed to support data access, analysis by end-users, and sometimes the maintenance of evidential value, as well as storage and transfer.

With respect to flexibility across platforms, while many of the listed formats support multiple platforms, tar requires third party tools on Windows, which may incur extra cost in terms of processing time and pose potential obstacles for long term preservation, as the third party tools are often not open source. One clear disadvantage of aff is that it assumes the image is from a disk as opposed to a collection of files or folders. However, this is not an insurmountable obstacle, as harvested websites can be, in theory, mounted on to virtual disks that are then turned into images using aff (see Figure 1). Further, an extension of aff, known as aff4, now allows the capture of webpages over the network as images. It may be too soon for aff4 to be employed as it may not be stable enough, but the format promises to be compatible with aff formats. This means a plan to use aff initially, with a view to migrate to aff4 when it becomes stable, is fully feasible.

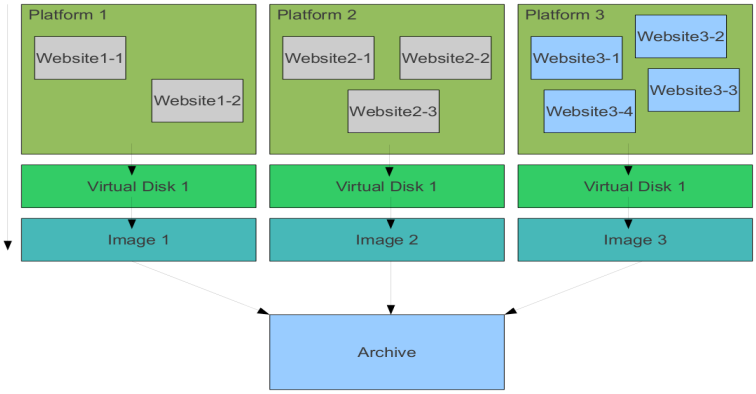

Figure 1. Workflow: Implementation of aff format using virtual disks.

In addition to what was mentioned above, the International Internet Preservation Consortium WARC format has been shown to have compatibility issues with the Internet Archive ARC format, even though it was created to accommodate previous data stored in the Internet Archive ARC format[20]. Data recovery problems have also been observed with respect to tar.[21]

[20] See: https://webarchive.jira.com/wiki/display/Heritrix/ARC+to+WARC+%28to+ARC%29

[21] See: http://www.linuxquestions.org/questions/linux-software-2/recovering-files-from-corrupt-tar-archive-326716/

 doi:10.2218/ijdc.v7i2.227

	tar	WARC	aff
Completeness	Partial File structure preserved but not other dependencies and change history.	No	Yes
Recoverability	No	Yes	Yes
In-Built Validation	Possible with gzip	No	Yes
Scalability	No Have to unpack everything before it can be searched or indexed. May have limits on size if it becomes huge.	Partial No information on whether it can be searched without unpacking and decompressing.	Yes
Transparency	Yes	Yes	Yes
Flexibility Embedding Metadata	No	Yes	Yes
Flexibility Handling Data	Partial Cannot control file sizes. Access possible using several software, but, software might be proprietary.	Partial Rendered accessed only by Internet archive software. As it does not interact with embedded data, size may be difficult to control	Partial Input data only in the form of disks. Easy manipulation of data chunk size. Access possible using several access software.

Table 2. Comparison of seven attributes across three formats, tar, WARC and aff.

Comparison of tar, WARC, and aff

In Table 2, we have summarised aspects of the seven attributes with respect to three file formats: tar, WARC, and aff. The description in Table 1 illustrates that:

1. The tar format has limited provisions for validation or recovery mechanisms, and no support for metadata. While the format allows working with various media types and collections, it does not allow user-defined block sizes. The format does retain file structure information and sometimes even file permissions, but it does not retain sector by sector information including unused space.
2. While WARC is specific to web crawls and therefore may provide features that are not available to other generic formats, the biggest drawback for this format is that rendered access is available only using the Internet Archive Way-Back Machine.

3. The Advanced Forensic File (aff) format is clearly the most robust in that it stores sector by sector information as a sequence of user-defined block sizes designed for maximum recovery when an error is found, has an in-built validation mechanism, and allows user-defined metadata.

Another attractive feature of the aff format is that the collection can be searched and indexed without decompression or unpacking. While the aff format is limited to imaging disks, we have already been pointed out that this can be partially circumvented with the use of virtual disks.

Conclusions

In this document, we made some observations on the advantages of employing forensic file formats (more specifically, the aff format) in a digital archive. We have:

1. Discussed attributes for file formats that need to be considered within an archive to support digital preservation;
2. Compared a broad range of file formats with respect to seven core file format attributes;
3. Made a direct comparison of three of the file formats, tar, WARC, and aff; and,
4. Proposed the Advanced Forensic File (aff) format, as the most robust among the three formats as a data-mining aware preservation storage format, where the preservation of a complex system of different file types is required – a situation often encountered within, but not limited to, a web archive.

While the aff format was originally intended for use in imaging disks (Garfinkel, 2006; Panda, Giordano & Kalil, 2006), we have illustrated that this limitation can be partially overcome through the use of virtual disk technology. Once the virtual disk technology is used to extend aff functionality, aff can be deployed as a storage container format for diverse types of media and information, such as tapes and data streams. In the context of information from the web crawled automatically, the virtual disk approach would not capture all the information available at the time of creation, which is often beyond our reach. However, it still helps us to work towards preserving the information we gather at the time of capture. This serves the purpose of not only supporting the preservation of the targeted information, but also recording the process by which we have gathered and processed the information, as the data capture history will be preserved in the aff disk image.

In digital forensics, the fidelity, integrity and authenticity of the data is crucial, as it directly links to the weight and sometimes even the admissibility of the object content as evidence in judicial settings (Goodin, 2011; Bell & Boddington, 2010). The forensics community is sensitive to the vital role of tracing data history. For example, the provenance of data and how the data was changed plays a part in understanding accountability and discovering evidence. The discipline's focus on not tampering with the data, even at the time of searching (e.g. no decompression and unpacking of the storage), is intended to ensure that the integrity of the digital material is maintained. As such, the handling of data within digital forensics is centred around preservation

 doi:10.2218/ijdc.v7i2.227

aims. Further, as forensics often involves making connections between several information entities, it is rapidly opening up to supporting data mining techniques (see Louis & Engelbrecht, 2011). The possibility of processing data in an archive without unpacking and decompressing reduces overheads in implementing these processes. It is also a valuable property with respect to basic large dataset indexing and search, which are must-preserve functionalities within the web data context. By absorbing digital forensics technology into the archival storage architecture, we could bring together the strengths of digital forensics that focuses on preserving digital information as evidence (data and interaction), and the wider context of preserving digital information, to introduce a preservation approach that also supports future data mining potential. The main questions to be answered to carry out the adoption of aff are: how will information be captured into virtual disks (e.g. will blogs from one website be kept together?), and how will the information within each object be segmented and distributed?

Next Steps

We suggest that a small-scale experiment be conducted to compare the formats tar, WARC and aff, (and possibly AXF format, which has not been properly examined here), using compression ratio, speed, and preservation attributes as evaluation criteria. The experiment should be based on a framework that can be used as a benchmark for comparing currently available container formats, as well as evaluating the suitability of new formats as they emerge. The steps of such an experiment must:

- Include the precise definition of the experimental context (e.g. research communities, public sector, business);
- Investigate the variance of performance with respect to the heterogeneity of data types (e.g. file types, programs, databases);
- Examine the scalability over a range of data collection sizes (say, from one gigabyte to ten terabytes); and,
- Compare the difficulties posed by equipment (e.g. processors, bandwidth, device type), and software constraints (e.g. operating systems).

In addition, it must also be emphasised that rigorous quantitative measures for each of the seven attributes should be developed so that each experiment can be replicated, compared, reviewed and validated within the information sciences community.

Acknowledgements

The research leading to the discussion in this paper was conducted as part of the BlogForever project funded by the European Union's Seventh Framework Programme (FP7-ICT-2009-6), under grant agreement number 269963.

References

Abrams, S. (2007). Instalment on file formats. In J. Davidson, K. Ashley, S. Ross, M. Day & F. Kennedy (Eds.), *Curation Reference Manual.* Digital Curation Centre. Retrieved from http://www.dcc.ac.uk/resources/curation-reference-manual/completed-chapters/file-formats

Arms, C. & Fleischhauer, C. (2003). Digital formats: Factors for sustainability, functionality, and quality. Paper presented at the IS& T Archiving Conference. Society for Imaging Science and Technology, Washington, DC. Retrieved from http://memory.loc.gov/ammem/techdocs/digform/Formats_IST05_paper.pdf

Batini, C. & Scannapieco, M. (2006). *Data quality: Concepts, methodologies and techniques.* Berlin, Germany: Springer.

Bell, G.B. & Boddington, R. (2010). Solid state drives: The beginning of the end for current practice in digital forensic recovery? *Journal of Digital Forensics, Security and Law, 5*(3). Retrieved from http://www.jdfsl.org/subscriptions/JDFSL-V5N3-Bell.pdf

Brown, A. (2008). *Digital preservation guidance note 1: Selecting file formats for long-term preservation.* UK National Archives. Retrieved from http://www.nationalarchives.gov.uk/documents/selecting-file-formats.pdf

Buckley, R. (2008). *Technology watch report: JPEG 2000 - A practical digital preservation standard?* Digital Preservation Coalition Technology Watch Series Report. Retrieved from http://www.dpconline.org/component/docman/doc_download/87-jpeg-2000-a-practical-digital-preservation-standard

Checkland, P. (1981). *Systems thinking, systems practice.* Chichester, UK: John Wiley & Sons.

Christensen, S.S. (2004). *Archival format and metadata requirements.* Report by the Royal Library, Copenhagen, Denmark, and the State and University Library, Århus, Denmark. Retrieved from http://netarkivet.dk/wp-content/uploads/Archival_format_requirements-2004.pdf

Collin, L. (2005). *A Quick Benchmark: Gzip vs. Bzip2 vs. LZMA*. Retrieved from http://tukaani.org/lzma/benchmarks.html

Dappert, A. & Farquhar, A. (2009). Significance is in the eye of the stakeholder. In M. Agosti et al. (Eds.), *European Conference on Digital Libraries (ECDL)*. Retrieved from http://www.bl.uk/aboutus/stratpolprog/ccare/pubs/2009/ipres2009-Dappert%20and%20Farquhar.pdf

Fanning, B.A. (2008). *Technology watch report: Preserving the data explosion - Using PDF.* Digital Preservation Coalition Technology Watch Series Report. Retrieved from http://www.dpconline.org/component/docman/doc_download/86-preserving-the-data-explosion-using-pdf

Garfinkel, S.L. (2006). AFF: A new format for storing hard drive images. *Communications of the ACM, 49*(2). Retrieved from http://cacm.acm.org/magazines/2006/2/6013/fulltext

Goodin, D. (2011). *Self-erasing flash drives destroy court evidence: Golden age of forensics coming to close*. The A Register. Retrieved from http://www.theregister.co.uk/2011/03/01/self_destructing_flash_drives/

Guttenbrunner, M., Wieners, J., Rauber, A. & Thaller, M. (2010). Same same but different: Comparing rendering environments for interactive digital objects. In M. Ioannides (Eds.), *EuroMed 2010*, *LNCS 6436*. Springer Verlag, Heidelberg. Retrieved from http://www.euromed2010.eu/e-proceedings/content/full/140.pdf

Hedstrom, M. & Lee, C.A. (2002). Significant properties of digital objects: definitions, applications, implications. *In Proceedings of the DLM-Forum 2002.* Retrieved from http://www.ils.unc.edu/callee/sigprops_dlm2002.pdf

Huc, C. et al. (2004). *Criteria for evaluating data formats in terms of their suitability for ensuring long term information preservation, v.5.* Groupe Pérennisation des Informations Numériques (PIN). Retrieved from http://www.docstoc.com/docs/23329435/criteria-for-evaluating-data-formats

Klausmann, T. (2008). *Gzip, Bzip2 and Lzma compared*. Retrieved from http://blog.i-no.de//archives/2008/05/08/index.html#e2008-05-08T16_35_13.txt

Louis, A.L. & Engelbrecht, A.P. (2011). Unsupervised discovery of relations for analysis of textual data. *Digital Investigations, 7.*

McLellan, E.P. (2006). *General study 11 final report: Selecting digital file formats for long-term preservation.* InterPARES 2 Project. Retrieved from http://www.interpares.org/display_file.cfm?doc=ip2_file_formats(complete).pdf

Nieminen, J. (2004). *Archiver comparison.* Retrieved from http://warp.povusers.org/ArchiverComparison/

Panda, B., Giordano, J. & Kalil, D. (2006). Next-generation cyber forensics. *Communications of the ACM, 49*(2). Retrieved from http://cacm.acm.org/magazines/2006/2/5997/fulltext

Pipino, L.L., Lee, Y.W. & Wang, R.Y. (2002). Data quality assessment. *Communications of the ACM, 45*(4). Retrieved from http://web.mit.edu/tdqm/www/tdqmpub/PipinoLeeWangCACMApr02.pdf

Rog, J. & van Wijk, C. (2008). *Evaluating file formats for long-term preservation.* Koninklijke Bibliotheek National Library of Netherlands. Retrieved from http://www.kb.nl/hrd/dd/dd_links_en_publicaties/publicaties/KB_file_format_evaluation_method_27022008.pdf

Ross, S. (2006). Approaching digital preservation holistically. In A. Tough and M. Moss (Eds.), *Information Management and Preservation.* Oxford, UK: Chandos Press.

Todd, M. (2009). *Technology watch report: File formats for preservation.* Digital Preservation Coalition Technology Watch Series Report. Retrieved from http://www.dpconline.org/component/docman/doc_download/375-file-formats-for-preservation

doi:10.2218/ijdc.v7i2.227

Appendix

Table 1. List of widely used storage container formats.

Reference Number	Acronym	Expansion	Developers	Description	URL
A1	aff	Advanced forensics format	Simson Garfinkel & Basis Technology	Extensible open format for the storage of disk images and related forensic metadata, using segments.	http://www.forensicswiki.org/wiki/AFF
A2	aff4	Advanced forensic framework 4	Michael Cohen, Simson Garfinkel, & Bradley Schatz	Evidence management system integrated within the file specification	http://www.forensicswiki.org/wiki/AFF4
A3	Arc (IA)	Internet archive archive format	Internet Archive (Mike Burner & Brewster Kahle)	Format of aggregate files. It must be possible to concatenate multiple archive files in a data stream.	http://www.archive.org/web/researcher/ArcFileFormat.php
A4	ARC (SEA)	System enhancement associates archive format	System Enhancement Associates (Thom Henderson)	Lossless data compression and archival format. Legacy format incapable of compressing entire directory trees.	http://www.fileformat.info/format/arc/corion.htm
A5	AXF	Archive exchange format	—	The AXF object contains the payload accompanied by structured or unstructured metadata, checksum and provenance information, full indexing structures in an encapsulated package.	http://www.openaxf.org/
A6	BagIt	—	California Digital Library	Storage and network transfer of arbitrary digital content, using file system directories. A “bag” consists of a “payload” (the arbitrary content) and “tags”, which are metadata files intended to document the storage and transfer of the bag.	http://tools.ietf.org/html/draft-kunze-bagit-06

Reference Number	Acronym	Expansion	Developers	Description	URL
A7	bzip2	—	Julian Seward	Lossless data compression algorithm that uses the Burrows–Wheeler transform to convert frequently-recurring character sequences into strings of identical letters.	http://bzip.org/
A8	cfs	Compact file set	Pismo Technic Inc.	Open archive file format and software distribution container file format. Mostly for reading optical media.	http://www.pismotechnic.com/cfs/
A9	cpio	Copies (cp) into or out of (io) archive	Originally Unix, later GNU version developed	Tape archiver as part of PWB/UNIX. Later developed into GNU cpio. Usually tar is now preferred.	http://www.gnu.org/software/cpio/cpio.html
A10	dd raw image	Disk duplication	Originally Unix, later made available on Linux distributions.	Raw sector-by-sector image data. No metadata data. No built-in compression.	http://linux.die.net/man/1/dd
A11	dmg	—	Apple Macintosh	MAC OS X disk imaging format.	Wikipedia article: http://en.wikipedia.org/wiki/Apple_Disk_Image
A12	EnCase image format	—	EnCase	Closed format used by EnCase based on ASR Data's Expert Witness Compression Format.	http://www.forensicswiki.org/wiki/EnCase
A13	gfzip	—	gfz project	Forensics File Format, allowing non-sequential access. Development frozen since 2006.	http://gfzip.nongnu.org/filespec.html
A14	gzip	Gnu zip	GNU project (Jean-Loup Gailly & Mark Adler)	Compression algorithm based on a combination of Lempel-Ziv (LZ77) and Huffman coding.	http://www.gzip.org/
A15	iso image	ISO 9660:1988, ECMA-119	—	Optical media disk imaging format.	http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=17505

doi:10.2218/ijdc.v7i2.227

Reference Number	Acronym	Expansion	Developers	Description	URL
A16	jar	Java archive	—	Format for aggregating java class file library.	http://download.oracle.com/javase/6/docs/technotes/guides/jar/jar.html
A17	kgb	KGB Archiver	Tomasz Pawlak	Compression and archiver based on the PAQ6 algorithm.	http://kgbarchiver.net/cgi-sys/suspendedpage.cgi (site suspended at the time of writing this paper)
A18	lzma	Lempel–Ziv–Markov chain algorithm	Igor Pavlov – some question whether Pavlov is the creator.	First used in the 7z format of 7-zip. Default compression method used in 7-zip.	http://www.7-zip.org/
A19	lzop	—	Markus F.X.J. Oberhumer	Lossless data compression library written in ANSI C that favours speed over compression ratio.	http://www.lzop.org
A20	partimage	Partition image	Francois Dupoux & Franck Ladurelle	Disk cloning utility for Linux/Unix for the purpose of recovery. Limited to supported file system types and does not clone unused portions.	http://www.partimage.org/
A21	PeaZip	—	PEAZIP SRL	File archiver for Windows and Linux.	http://www.peazip.org/
A22	rar	Roshal ARchive	Eugene Roshal	Proprietary compression utility with a closed algorithm. Owned by Alexander L. Roshal.	http://www.rarlab.com/
A23	7-zip	—	Igor Pavlov	7-zip is a utility with native archiving format 7z which uses the lzma compression algorithm.	http://www.7-zip.org/
A24	sgzip	—	Australian Department of Defence	Native forensics file format for PyFlag.	http://www.forensicswiki.org/wiki/Pyflag

Reference Number	Acronym	Expansion	Developers	Description	URL
A25	shar	Shell archive	Unix	This is a utility for creating a shell script. Running the script will recreate the files. Currently tar is preferred because executables pose risk to the system. Related to GNU Sharutils.	http://linux.die.net/man/1/shar
A26	tar	Tape archive format	Originally Unix command. Later developed into GNU versions.	The format was created for tape backup purposes in the early days of Unix and standardized by POSIX.1-1988 and later POSIX.1-2001. Later developed into the widely distributed GNU tar.	http://www.gnu.org/software/tar/
A27	WARC	Web archive format	International Internet Preservation Consortium	Next generation (taking after Internet archive's Arc format) aggregated file format.	http://archive-access.sourceforge.net/warc/
A28	xz-utils	—	The Tukaani Project	Free compression software including LZMA and xz for UNIX-like operating systems.	http://tukaani.org/xz/
A29	zip	Originally coined to convey "speed"	Phil Katz	Created to replace ARC by System Enhancement Associates (see above). Originally part of PKZIP for Microsoft Windows.	http://www.pkware.com/documents/casestudies/APPNOTE.TXT